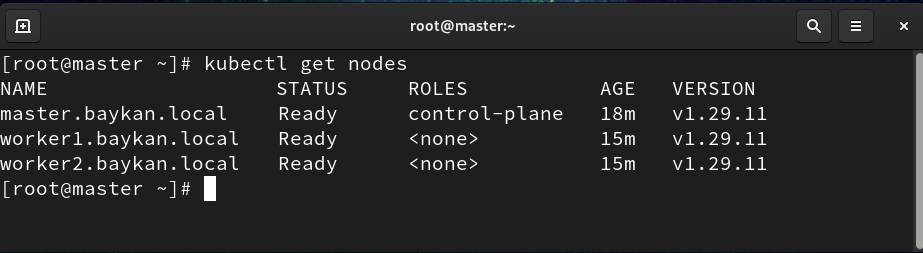
|  |  |
| --- | --- |
| **DevOps Project** | |
| **Project Name** | Deploy on Kubernetes |
| **Project Engineer** | Zeynep BAYKAN |

**Project Purpose**

To perform deployment processes through ArgoCD using Kubernetes infrastructure.

**Objectives**

Planning, setting up, and managing deployment processes for Kubernetes infrastructure with a Cluster/Load Balancer on Public Cloud platforms (Azure, AWS, Google Cloud). This setup will consist of three nodes: one master node and two worker nodes.

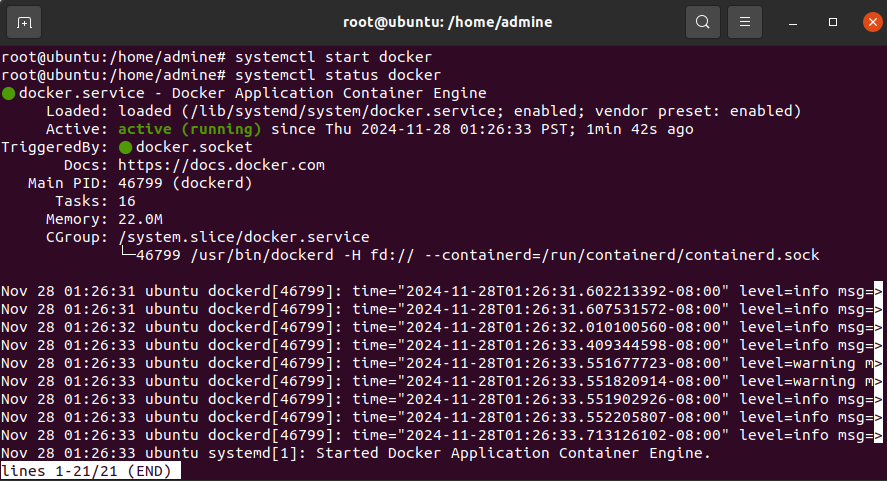


Deployments will be carried out through ArgoCD, which will be installed on this structure. Before this work, the project was minimized and created in a local VmWare virtualization environment, and its tests have been conducted. The details of this work are presented below for your information. My personal goal is to rapidly elevate my technical knowledge to a senior level and perform deployments on a Public Cloud or Dedicated Server hosted in a Data Center.

**Installation**

All installation processes have been carried out on a VM with the Ubuntu-20.04 operating system installed on Vmware. The Kubernetes infrastructure for the test environment was set up using MiniKube on this VM. All installation and testing processes:

1. Docker Engine Installation

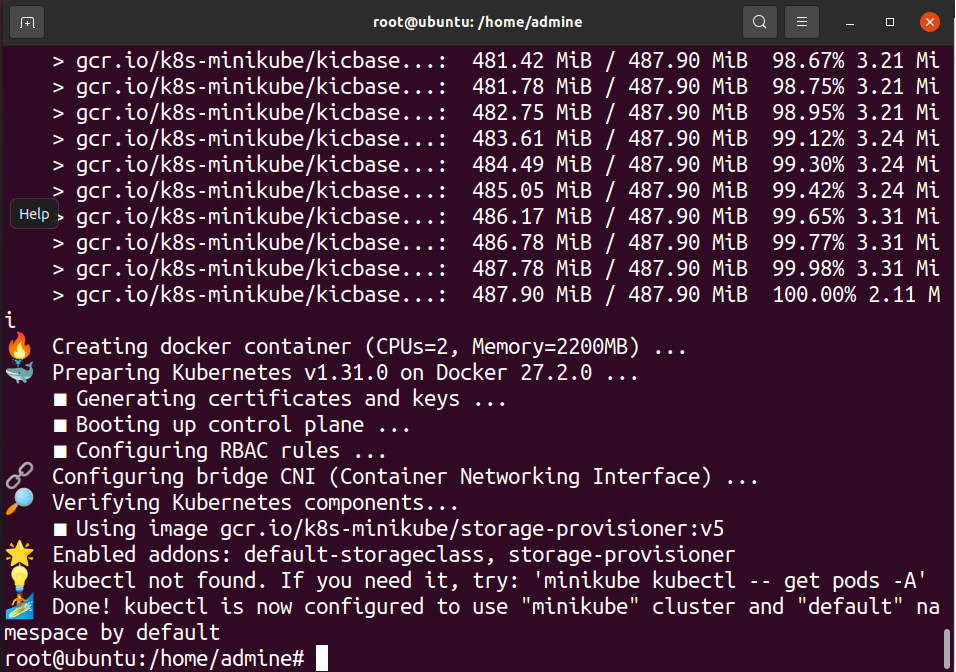
 After Docker Engine was installed on the VM, the Docker service was enabled and started.

2.Kubernetes Installation

In the test environment, Kubernetes was installed as "root" using Minikube, and the service was started to initiate Minikube.

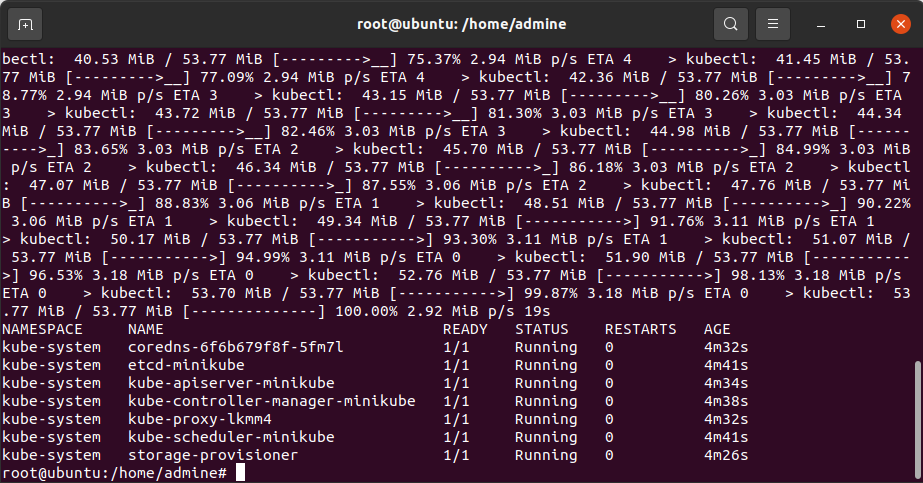
Command:

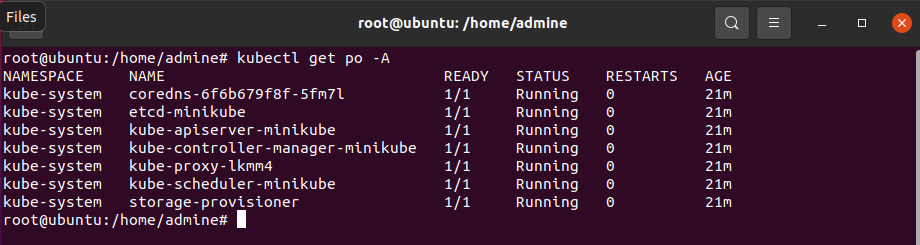
minikube start



A cluster was created for Minikube.To view the pods across all namespaces, the following command was used:

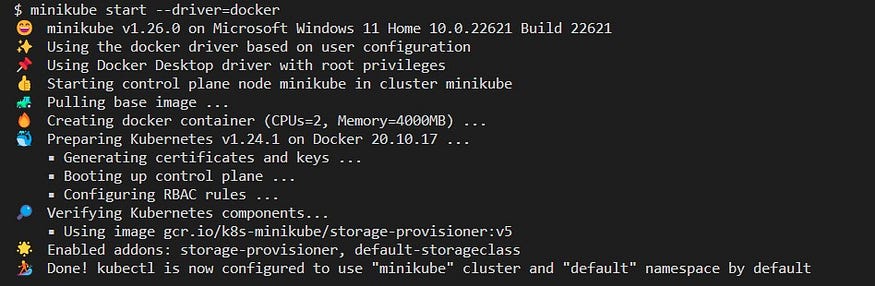
minikube kubectl -- get po -A





3.ArgoCD Installation

After setting up the Kubernetes cluster with Minikube, ArgoCD was installed for deployment operations.

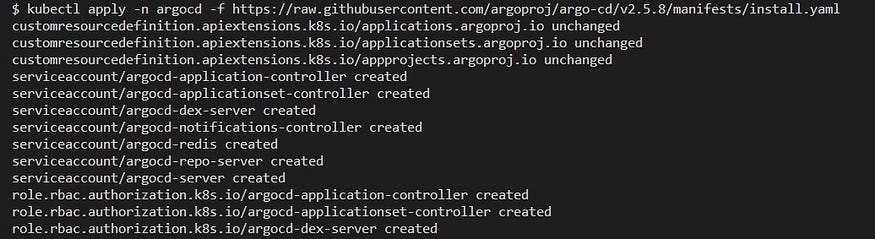
minikube start --driver=docker

The necessary namespace for ArgoCD was defined:

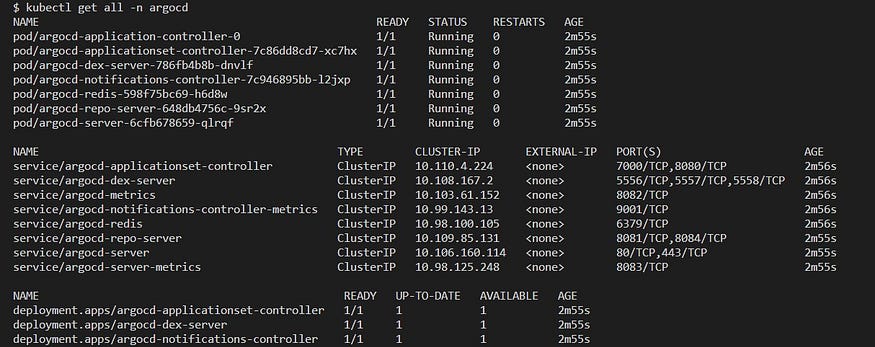
kubectl create ns argocd

The installation files were then used from GitHub.

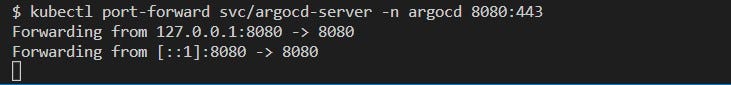
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/v2.5.8/manifests/install.yaml



All objects within the ArgoCD namespace were checked using the following command:



ArgoCD management consists of a web interface dashboard. After ensuring that the pods are running, port forwarding was set up to access this web interface.

kubectl port-forward svc/argocd-server -n argocd 8080:443

Access to ArgoCD is provided through the browser by navigating to localhost:8080.

